under test (DUT);

a test result compaction module composer to generate a test result compaction module code; and

a code merger to merge code from the RIT-G composer, the test execution directive composer and the test result compaction module composer to generate the software built-in self-test engine (SBE).

A1 Concl.

3. (Amended) The system as claimed in claim 1, wherein said SBE is to be merged with an expected test result and then loaded on-board a complex device under test (DUT) so as to activate a re-generative functional test on the complex device under test (DUT) and make a comparison between test results of the re-generative functional test and the expected test result to check for design validations and/or manufacturing defects.

A2

5. (Amended) The system as claimed in claim 2, wherein said SBE generation tool is a software tool installed to generate the software built-in self-test engine (SBE), and wherein individual components of said SBE generation tool, including the random instruction test generator (RIT-G) composer, the test execution directive composer, the test result compaction module composer, and the code merger, are software modules written in any computer language.

A3

7. (Amended) The system as claimed in claim 2, wherein said SBE generation tool is a hardware implementation installed to generate the software built-in self-test engine (SBE).

8. (Amended) The system as claimed in claim 2, wherein said run time environment

includes a test execution environment including an exception handler to handle illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops, and a RIT environment to provide equivalent operating system (OS) functions needed by the RIT generator to generate the re-generative functional test.

9. (Amended) The system as claimed in claim 2, wherein said compact RIT-G code produced is a C-language program compiled by a C-compiler to produce an assembly language version of the RIT-G code, and when the run time environment, the test result compaction module code and the assembly language version of the RIT-G code are assembled by an assembler, a single program indicating the SBE in the target DUT's object code is obtained.

A3 cont.

10. (Amended) The system as claimed in claim 9, wherein said compact RIT-G code includes an instruction generation module to generate individual instructions during testing application.

11. (Amended) The system as claimed in claim 1, wherein said software built-in self-test engine (SBE) comprises:

a RIT generator including RIT machine code reside on-board the complex device under test (DUT) for generating the re-generated functional test;

a test program execution module including test execution directives for providing a run time environment to store and run the re-generated functional test; and

a test result compaction module including compression machine code to compress test results of the re-generated functional test for storage on-board the complex device under test (DUT).

12. (Amended) The system as claimed in claim 11, wherein said test execution environment employs an exception handler to handle illegal conditions, including undesirable memory accesses, deadlock, shut-down, and infinite loops.

A3 cont'd

13. (Amended) The system as claimed in claim 1, wherein said complex device under test (DUT) includes a microprocessor.

A4

18. (Amended) The computer readable medium as claimed in claim 17, wherein said SBE generation tool comprises:

a random instruction test generator (RIT-G) composer to receive the user directives and the instruction information and generate a compact RIT-G code;

a test execution directive composer to receive the user directives and the device constraints and create a run time environment needed to enable the re-generative functional test to repeatedly generate functional tests and execute generated tests on-board the complex device under test (DUT);

a test result compaction module composer to generate a test result compaction module code; and

a code merger to merge code from the RIT-G composer, the test execution directive composer and the test result compaction module composer to generate the software built-in self-test engine (SBE).

19. (Amended) The computer readable medium as claimed in claim 18, A4 cont'd

wherein said SBE is to be merged with an expected test result and then loaded on-board a complex device under test (DUT) so as to activate a re-generative functional test on the complex device under test (DUT) and make a comparison between test results of the re-generative functional test and the expected test result to check for design validations and/or manufacturing defects.

A5

21. The computer readable medium as claimed in claim 18, wherein said run time environment includes a test execution environment including an exception handler to handle illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops, and a RIT environment to provide equivalent operating system (OS) functions needed by the RIT generator to generate the re-generative functional test.

22. (Amended) The computer readable medium as claimed in claim 18, wherein said compact RIT-G code produced is a C-language program compiled by a C-compiler to produce an assembly language version of the RIT-G code, and when the run time environment, the test result compaction module code and the assembly language version of the RIT-G code are assembled by an assembler, a single program indicating the SBE in the target DUT's object code is obtained.

23. (Amended) The computer readable medium as claimed in claim 18, wherein said compact RIT-G code includes an instruction generation module to generate individual instructions during testing application.

24. (Amended) The computer readable medium as claimed in claim 17, wherein said software built-in self-test engine (SBE) comprises:

a RIT generator including compact RIT machine code reside on-board the complex device under test (DUT) for generating the re-generated functional test;

A5 cont.

a test program execution module including test execution directives for providing a run time environment to store and run the re-generated functional test; and

a test result compaction module including compression machine code to compress test results of the re-generated functional test for storage on-board the complex device under test (DUT).

28. (Amended) The method as claimed in claim 27, wherein said software built-in self-test engine (SBE) is generated by:

A6

generating a compact random instruction test generator (RIT-G) code based on the user directives and the instruction information;

creating a run time environment to enable the re-generative functional test to repeatedly generate functional tests and execute generated tests on-board the complex device under test (DUT) based on the device constraints;

generating a test result compaction module code based on the user directives and the device constraints; and

merging the RIT-G code, the run time environment and the test result compaction module code to obtain the software built-in self-test engine (SBE).

29. (Amended) The method as claimed in claim 27, wherein said SBE is to be merged with an expected test result and then loaded on-board a complex device under test (DUT) so as to activate a re-generative functional test on the complex device under test

A6 concld

(DUT) and make a comparison between test results of the re-generative functional test and the expected test result to check for design validations and/or manufacturing defects.

A7

31. (Amended) The method as claimed in claim 28, wherein said run time environment includes a test execution environment including an exception handler to handle illegal conditions such as undesirable memory accesses, deadlock, shut-down, and infinite loops, and a RIT environment to provide equivalent operating system (OS) functions needed by the RIT generator to generate the re-generative functional test.

32. (Amended) The method as claimed in claim 28, wherein said compact RIT-G code produced is a C-language program compiled by a C-compiler to produce an assembly language version of the RIT-G code, and when the run time environment, the test result compaction module code and the assembly language version of the RIT-G code are assembled by an assembler, a single program indicating the SBE in the target DUT's object code is obtained.

33. (Amended) The method as claimed in claim 28, wherein said complex device under test (DUT) include a microprocessor.